

Міністерство освіти і науки України
Національний університет водного господарства та
природокористування
Кафедра гідроінформатики

01-02-162

Методичні вказівки

до практичних занять та самостійної роботи з навчальної дисципліни
«Мова програмування і програмне середовище R у водній інженерії» для здобувачів вищої освіти першого (бакалаврського) рівня усіх освітньо-професійних програм спеціальностей НУВГП денної форми навчання

Схвалено науково-методичною радою
НУВГП протокол № 2 від 26.03.2020 р.

Методичні вказівки до практичних занять та самостійної роботи з навчальної дисципліни «Мова програмування і програмне середовище R у водній інженерії» для здобувачів вищої освіти першого (бакалаврського) рівня усіх освітньо-професійних програм спеціальностей НУВГП денної форми навчання [Електронне видання] / Новачок О. М. – Рівне : НУВГП, 2020. – 24 с.

Укладач: Новачок О. М., кандидат сільськогосподарських наук, доцент кафедри гідроінформатики.

Відповідальний за випуск: Клімов С. В., кандидат технічних наук, доцент, завідувач кафедри гідроінформатики.

Вчений секретар
науково-методичної ради

Костюкова Т. А.

Зміст

Вступ	3
1. Знайомство з R.....	4
2. Об'єкти і типи даних в R	8
3. Функції і конструкції мови R. Статистика	10
4. Графіка і графічні параметри	15
5. Доступ до віддалених баз даних водогосподарського призначення.....	20
Рекомендована література	24

© О. М. Новачок, 2020
© НУВГП, 2020

Вступ

Методичні вказівки призначені для використання при вивченні дисципліни «Мова програмування і програмне середовище R у водній інженерії» студентами вільного вибору всіх спеціальностей денної форми навчання.

Метою навчальної дисципліни є оволодіння студентами сучасними методами та засобами прийняття інженерних рішень у водогосподарській галузі на засадах математичного моделювання та комп'ютерних технологій з використанням вільного програмного забезпечення; формування у студентів системного, аналітичного мислення для оцінки ситуацій, що виникають.

Основним завданням вивчення дисципліни «Мова програмування і програмне середовище R у водній інженерії» є:

- ознайомлення студентів з можливостями сучасного вільного програмного забезпечення з метою його використання у водогосподарській галузі;
- розкриття можливостей ефективного застосування інформаційних технологій у водній інженерії.

Після вивчення дисципліни студенти повинні **знати**:

- особливості вільного програмного забезпечення;
- сучасні інформаційні технології у водному господарстві, вимоги до них;
- системи для високоякісного представлення даних в графічному вигляді.

Після вивчення дисципліни студенти повинні **вміти**:

- використовувати сучасне вільне програмне забезпечення (мову програмування R) у водній інженерії, проводити необхідні оптимізаційні розрахунки;
- використовувати, джерела публічної інформації, вільне програмне забезпечення для вирішення водогосподарських проблем.

Проблема неліцензійного програмного забезпечення сьогодні стоїть особливо гостро. І якщо раніше тези про захист авторських прав на комп'ютерні програми закріплювались тільки на папері, то сьогодні вони знаходять своє втілення в реальному житті у вигляді перевірок контролюючими органами, штрафів, вилучення комп'ютерів і судових розглядів.

Цивільно-правова відповідальність:

Згідно ст. 432 Цивільного кодексу України правовласники програмного забезпечення мають право подати позов до судових органів у разі порушення їх авторського права. Можна виділити наступні можливі рішення суду (ч. 2 ст. 52 Закону України «Про авторське право і суміжні права»):

відшкодування моральної та матеріальної шкоди; стягнення з порушника доходів, отриманих в наслідок порушення; опублікування в ЗМІ інформації про порушення; виплата компенсації в розмірі **від 10 до 50 000 мінімальних зарплат** і ін.

1. Знайомство з R

R - мова програмування і середовище вільного програмного забезпечення для статистичних обчислень і графіки, що підтримуються R Foundation for Statistical Computing. Мова R широко використовується для статистики та інтелектуального аналізу даних, для розробки статистичного програмного забезпечення. Опитування, дослідження інтелектуального аналізу даних і дослідження літератури наукових баз даних показують значне збільшення популярності в останні роки. Станом на березень 2019 р. R посідає 14 місце в індексі TIOBE (міра популярності мов програмування).

Можливості R розширюються через створювані користувачем пакети, які дозволяють спеціалізовані статистичні методи, графічні пристрої, можливості імпорту та експорту, інструменти звітності. Ці пакети розробляються в основному в R, а іноді і в Java, C, C++ і Fortran.

Основний набір пакетів включено при встановленні R, з більш ніж 15 000 додаткових пакетів (станом на вересень 2018), доступних у Всеохопній мережі архівів Comprehensive R Archive Network (CRAN), Bioconductor, Omegahat, GitHub та інших.

Сторінка "Перегляди завдань" на веб-сайті CRAN перелічує широкий спектр завдань для яких доступні пакети R (у таких галузях, як фінанси, генетика, високопродуктивні обчислення, машинне навчання, медичні зображення, соціальні науки та просторова статистика). R також був визначений FDA як придатний для інтерпретації даних клінічних досліджень.

Інші ресурси R-пакета включають Crantastic, сайт спільноти для оцінки та перегляду всіх CRAN-пакетів, R-Forge, центральну платформу для спільної розробки пакетів R, R-програм і проектів. У R-Forge також розміщено багато неопублікованих бета-пакунків, а також розробки версій пакетів CRAN.

Розробка **R** відбувалась під істотним впливом двох наявних мов програмування: мови програмування S з семантикою успадкованою від Scheme. R названа за першою літерою імен її засновників Роса Іхаки (Ross Ihaka) та Роберта Джентлмена (Robert Gentleman) працівників Оклендського Університету в Новій Зеландії.

Незважаючи на деякі принципові відмінності, більшість програм, написаних мовою програмування S запускаються в середовищі R. R розповсюджується безкоштовно за ліцензією GNU General Public License у вигляді вільнодоступного вихідного коду або відкомпільованих бінарних версій більшості операційних систем: Linux, FreeBSD, Microsoft Windows, Mac OS X, Solaris. R використовує текстовий інтерфейс, однак існують різні графічні інтерфейси користувача. R має значні можливості для здійснення

статистичних аналізів, включаючи лінійну і нелінійну регресію, класичні статистичні тести, аналіз часових рядів (серій), кластерний аналіз і багато іншого. R легко розширюється завдяки використанню додаткових функцій і пакетів доступних на сайті Comprehensive R Archive Network (CRAN). Більша частина стандартних функцій R, написана мовою R, однак існує можливість підключати код написаний C, C++, або Фортраном. Також за допомогою програмного коду на C, C++, Java, .NET або Python можна безпосередньо маніпулювати R об'єктами.

Завдяки своїй S спадщині, R має більш сильне об'єктно-орієнтоване програмування об'єктів, ніж більшість статистичних мов обчислень. Розширення R також полегшується завдяки її лексичній області видимості правил. Іншою силою R є статичні графіки, які можуть виробляти графіки типографської якості, в тому числі математичні символи. Динамічні та інтерактивні графіки доступні через додаткові пакети. R має Rd, свій власний LaTeX формат документації, яка використовується для забезпечення вичерпної документації, як онлайн в різних форматах, так і в друкованому вигляді.

R належить до інтерпретованих мов програмування і для роботи використовується командний інтерпретатор.

R підтримує концепцію об'єктно-орієнтованого програмування (ООП) включаючи генеріс функції, результат виконання яких залежить від аргументів (типу об'єктів), що передаються генеріс функції. В мові програмування R всі змінні є об'єктами, кожен об'єкт належить до певного класу. При цьому R має дві класові моделі: S3 та S4. Перша була реалізована від початку існування R, друга була додана у версії 1.7.0 з пакетом methods. S3 не є справжньою класовою системою, класи S3-об'єкта визначаються простим атрибутом - вектором символьних рядків. При цьому, при виконанні генеріс функцій, таких як plot() чи summary(), диспетчер методів шукає в таблиці методів метод, який узгоджується з іменем першого аргумента.

Хоча R орієнтована на розв'язок і аналіз статистичних задач, вона може використовуватися для матричних обчислень з порівняльною швидкістю до математичних пакетів GNU Octave або MATLAB. Створено багато пакетів для статистичних обчислень, біоінформатики, оптимізації тощо. Середовище R містить засоби для візуалізації результатів обчислень (двовимірні, тривимірні графіки, діаграми, гістограми, діаграми (схеми) Ганта тощо). Графічні можливості R дозволяють створювати високоякісні графіки з різними атрибутами, зокрема математичні формули і символи. Іншою особливістю є функція Sweave яка дозволяє інтеграцію і виконання коду R в документах написаних за допомогою LaTeX з метою створення динамічних звітів. R de-facto став стандартом у міжнародній спільноті спеціалістів в галузі статистики, і широко використовується в розробках статистичних програм та аналізі даних. Згідно щорічного опитування REXTER's Annual Data

Miner Survey в 2010 році, більшість (43%) серед опитаних спеціалістів з аналізу даних використовують у своїй роботі середовище R.

Можливості R значно розширюються додатковими пакетами (бібліотеками). Пакети розробляються безпосередньо користувачами R. Існує понад 15000 пакетів, доступних на сайті Comprehensive R Archive Network (CRAN), Omegahat, Bioconductor, R-Forge. На сторінці "Task View" веб-сайту CRAN розміщено список напрямків (Фінанси, Генетика, Хеміометрія і Математична Фізика, Навколишнє середовище, Суспільні науки) в яких використовується R і для яких доступні пакети на сайті.

Для роботи з R існує кілька графічних інтерфейсів (GUI). Графічна оболонка RGui разом з командною оболонкою (терміналом) R Console входять до базового пакету R у версії для Windows.

RStudio — зручне кросплатформне середовище розробки з відкритим кодом (існує можливість запуску на віддаленому linux сервері).

RKward — розширюване середовище розробки IDE

RapidMiner і розширення RapidMiner R — середовище розробки для аналізу і обробки даних з використанням R.

WEKA Java Gui for R (JGR) — кросплатформний термінал і редактор R написаний на Java Deducer — графічний інтерфейс для аналізів даних з використанням системи меню (подібний до SPSS). Розроблений для використання разом з JGR та RGui.

Rattle GUI — кросплатформний графічний інтерфейс, розроблений для добування даних (збору та аналізу даних).

R Commander — кросплатформний GUI з системою меню і доступними додатковими плагінами (базується Tcl/Tk).

RExcel — додаток до Microsoft Excel, який дозволяє використовувати можливості R.

Sage — середовище для математичних розрахунків з використанням інтерфейсу веб-браузера, бібліотек R і підтримкою гру.

Red-R — інтерфейс для аналізу, що використовує R.

Tinn-R — графічний інтерфейс.

Текстові редактори та середовища розробки (IDE) з частковою підтримкою R: gedit, Bluefish, IDE Eclipse, Kate, Vim, Emacs (Emacs Speaks Statistics), Crimson Editor, ConTEXT, Tinn-R[15], Geany, jEdit, Syn, TextMate — The Missing Editor for Mac OS X, SciTE, WinEdt (R Package RWinEdt), WPE, notepad++ і SciViews.

R доступна для використання у мовах програмування Python (за допомогою пакета RPy), Perl (за допомогою модуля Statistics::R) і Ruby (за допомогою RSRuby).

Деякі пропріетарні програмні продукти призначені для аналізу статистичних даних (напр. SPSS, STATISTICA, SAS) мають розширення, розроблені для інтеграції у свої структури функціоналу R. Заснована 2007 року компанія Revolution Analytics розпочала комерційну підтримку версії R

під назвою ParallelR, розробленої спеціально для кластерів робочих станцій. В 2011 з'явилася можливість зчитувати і записувати дані у формат файлів SAS за допомогою пропріетарного Enterprise R.

Встановлення R в операційній системі Ubuntu. Початок роботи з R. Ознайомлення з основними можливостями. Завантаження даних та збереження результатів. R як калькулятор. Інсталяція та підключення додаткових пакетів в R.

Процедура встановлення мови програмування R описана на сайті <https://cran.r-project.org/> в розділі «R Binaries» підрозділі «linux/» підпідрозділі «ubuntu/». Для цього одночасно натискаємо три клавіші «Ctrl+Alt+t». Відкривається вікно терміналу (в операційній системі linux основні налаштування системи відбуваються саме в терміналі). Нова збірка ubuntu виходить двічі на рік – в квітні та жовтні. Збірка з довгим терміном підтримки (4 роки) має в назві LTS. Кожна збірка має умовну назву. Наприклад Ubuntu 16.04 (2016 рік 04 місяць) LTS має назву Xenial. В текстовому файлі /etc/apt/sources.list знаходиться перелік джерел (сайтів звідки встановлюються і поновлюються програми). Щоб додати до цього файлу джерела для R, в терміналі відкриваємо цей файл для редагування.

```
sudo gedit /etc/apt/sources.list
```

В цьому рядку записано, що Ви відкриваєте в редакторі gedit файл /etc/apt/sources.list з правами адміністратора (root, super user). Система попросить ввести пароль адміністратора. Копіюємо в цей файл рядок з сайту:

```
deb https://cloud.r-project.org/bin/linux/ubuntu/xenial-cran35/
```

Натискаємо дві клавіші «Ctrl+s», тобто зберігаємо зміни в файлі. Закриваємо файл. Тепер система «знає» звідки встановлювати мову R.

З сайту в термінал по-черзі копіюємо рядки. Щоб вставити рядок в термінал натискаємо праву кнопку миші у вікні терміналу і вибираємо «вставити».

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
E298A3A825C0D65DFD57CBB651716619E084DAB9
```

Цим ми встановлюємо ключ доступу до джерела R.

```
sudo apt-get update
```

Цим ми оновлюємо список джерел та перевіряємо наявність новіших версій програмного забезпечення які встановлені в системі.

```
sudo apt-get install r-base
```

Спостерігаємо за процесом встановлення базового комплекту мови програмування R.

В терміналі набираємо R і бачимо щось подібне;

```
nom@nom-2018:~$ R
```

```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
```

```
Copyright (C) 2020 The R Foundation for Statistical Computing
```

```
Platform: x86_64-pc-linux-gnu (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
```

You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

Система готова до роботи. Вона може працювати як калькулятор:

> 2+2

[1] 4

Результат в нижньому рядку.

> pi

[1] 3.141593

За замовчуванням в R встановлюється велика кількість базових функцій. Станом на 24.03.2020 року є 15478 додаткових пакетів, які мають детальні інструкції з їх встановлення та використання. Коли потрібно встановити додатковий пакет, наприклад “ggplot2” потрібно виконати наступну команду:

> install.packages(“ggplot2”)

і почекати поки він встановиться.

Перед використанням пакету потрібно виконати команду:

library(ggplot2)

Щоб вийти з системи R потрібно набрати:

q() попередні результати можна зберегти до наступного разу коли при виході натиснути «у».

Література: 1, 2, 3, 5.

2. Об’єкти і типи даних в R

Користувач може вибрати робочу директорію скориставшись функціями `getwd()` та `setwd("директорія")`. Для того, щоб уся зроблена робота не пропадала і можна було працювати над декількома проектами одночасно в R є можливість зберегти історію команд функцією `savehistory()` та весь поточний робочий простір `save.image()`. По замовчуванню, якщо не вказувати назви файлів, у робочій директорії будуть створені файли `.Rhistory` та `.RData`.

Вектори - це числовий масив, що може містити числові, текстові або логічні значення. Причому вектори повинні бути однотипними, тобто не можна змішувати у векторі дані різних типів як то числа і текст.

Створення числового і символьного векторів

> # Все, що за символом #, інтерпретується як коментар

> x <- c(1,2,3,4,5,6,7,8,9,10) # Створення числового вектора

> y <- 2^x # піднесення числа до степеня x


```

> y                                # перегляд змісту об'єкта y, аналогічно print(y)
[1]  2   4   8  16  32  64 128 256 512 1024

> b1 <- c("Kharkiv","Kyiv","Lviv") # символний вектор
> b1
[1] "Kharkiv" "Kyiv"  "Lviv"

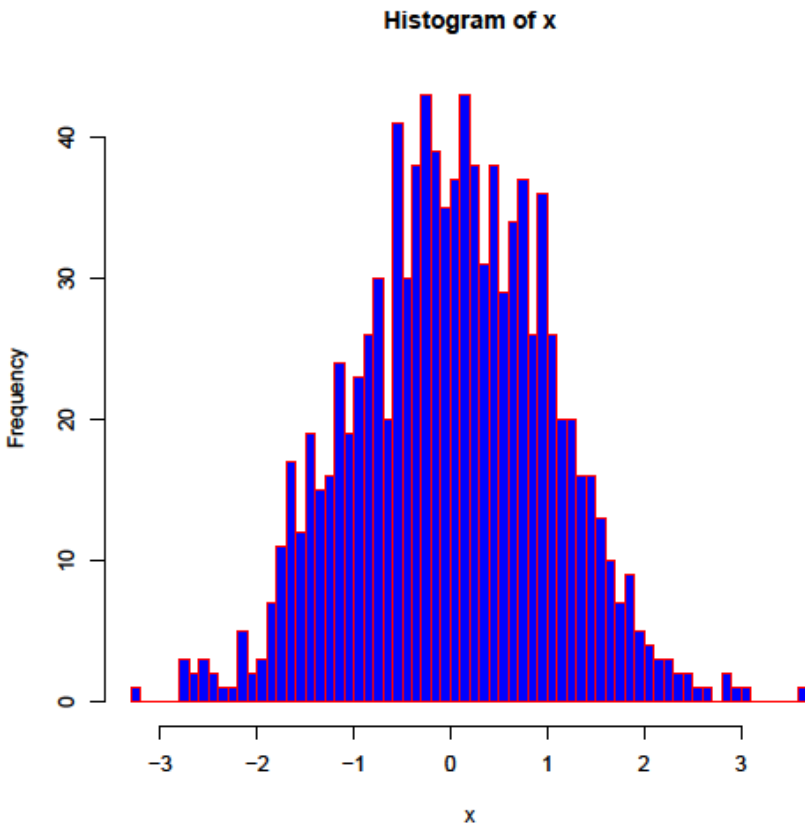
```

Генерація випадкових чисел нормального розподілу і побудова гістограми

```

> x <- rnorm(1000) # генерація 1000 випадкових чисел з розподілу Гауса
> histogram <- hist(x, breaks=50, plot=FALSE) # розрахунок гістограми для
змінної x, кількість інтервалів 50
> plot(histogram, col="blue",border="red") # зображення гістограми за
допомогою функції plot()

```



Матриці - це двовимірні масиви однотипних елементів. Припустимо, ми хочемо створити найпростішу матрицю 2×2 . Для початку створимо її з числового вектора:

```
> m <- 1:4
> m
[1] 1 2 3 4
> ma <- matrix(m, ncol=2, byrow=TRUE)
> ma
     [,1] [,2]
[1,]    1    2
[2,]    3    4
> str(ma)
int [1:2, 1:2] 1 3 2 4
> str(m)
int [1:4] 1 2 3 4
```

Масиви в R по суті ті ж матриці тільки більшої розмірності. Створюються масиви за допомогою функції `array()`.

Фрейми даних (data frame, також перекладають як таблиці даних) - це найзастосовуваніша структура даних. На відміну від матриць фрейми можуть містити елементи різних типів, що зручно для роботи з реальними даними. Створюється фрейм даних наступним чином:

```
mydata<-data.frame(col1, col2,col3,...)
```

де `col1`, `col2`, `col3` - це вектори будь-якого типу, що стануть стовпцями фрейму.

Фактори. Часто буває, що певні значення представлені не у чисельному вигляді (кількісні дані), а у вигляді певних текстових позначень (категоріальні дані, також називаються номінальними, неметричними або ж якісними). Існує можливість призначити їм додаткові чисельні значення, що в подальшому дає можливість порівнювати їх. Наприклад, якщо описується стан певного об'єкту через значення "прекрасно", "добре", "задовільно", "незадовільно", то можна призначити їм відповідні чисельні значення 4, 3, 2, 1. Для цього існує функція `factor()`. Функція може присвоювати значення в алфавітному порядку, або ж у заданому користувачем порядку.

Списки - це впорядкований набір об'єктів, що об'єднані під одним іменем. Причому об'єкти можуть бути різні, не зв'язані між собою. Для створення об'єктів використовується функція `list`.

```
mylist <- list(об'єкт 1, об'єкт2, ...).
```

Література: 1, 2, 3, 5.

3. Функції і конструкції мови R. Статистика

<code>help.start()</code>	#загальна довідка
<code>help("дещо")</code>	#допомога по функції "дещо" (лапки не обов'язкові)
<code>?дещо</code>	# теж саме

```

help.search("щось") #пошук записів, що містять "дещо"
??дещо             # теж саме
example("дещо")    # приклади використання
RSiteSearch("дещо") # пошук в інтернеті
apropos("дещо", mode="function") #список всіх доступних функцій, в назві
яких є "дещо"
data()             # список всіх демонстраційних даних, що містяться в
завантажених пакетах
vignette()         # список всіх доступних посібників по завантажених
пакетах
vignette("дещо")   # список посібників по темі "дещо"

```

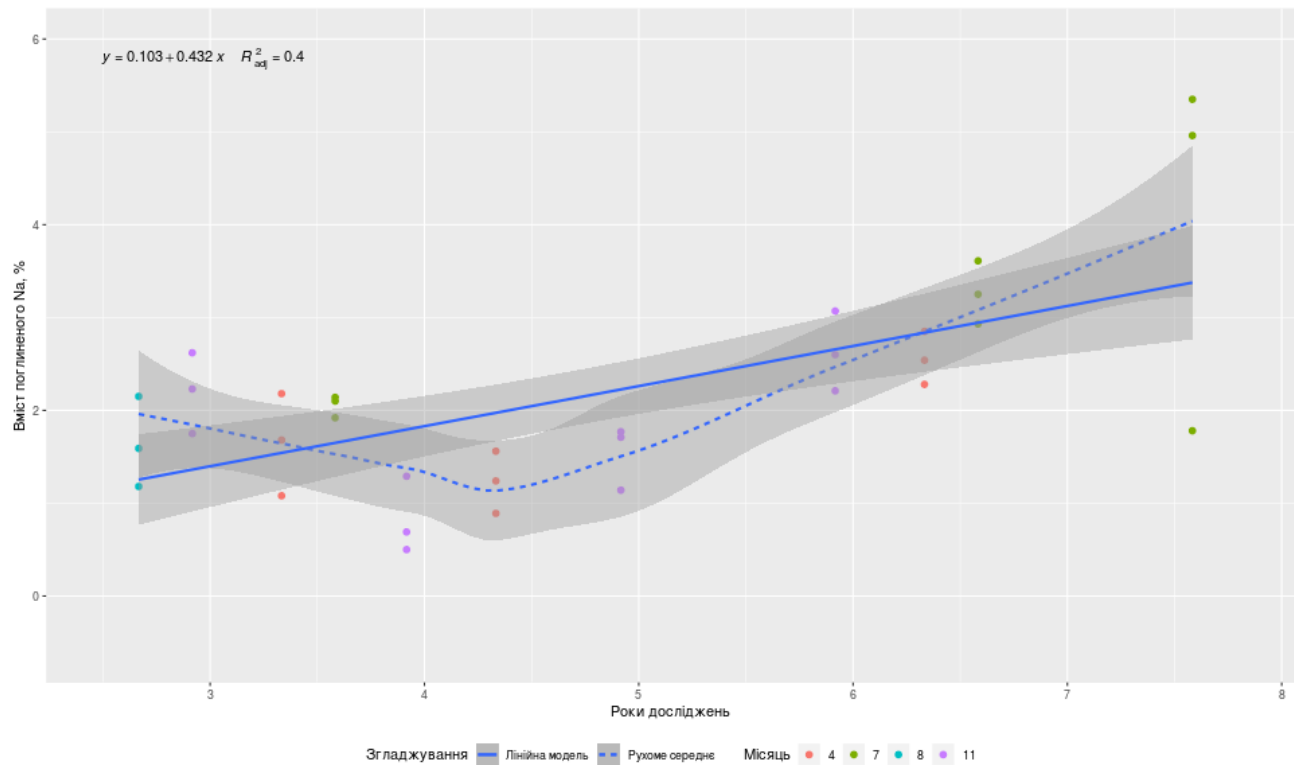
R має розвинуту систему допомоги. Допомога викликається `help()`, де в дужках потрібно вказати тему. Наприклад, щоб дізнатись як вийти з системи потрібно набрати `help(q)`.

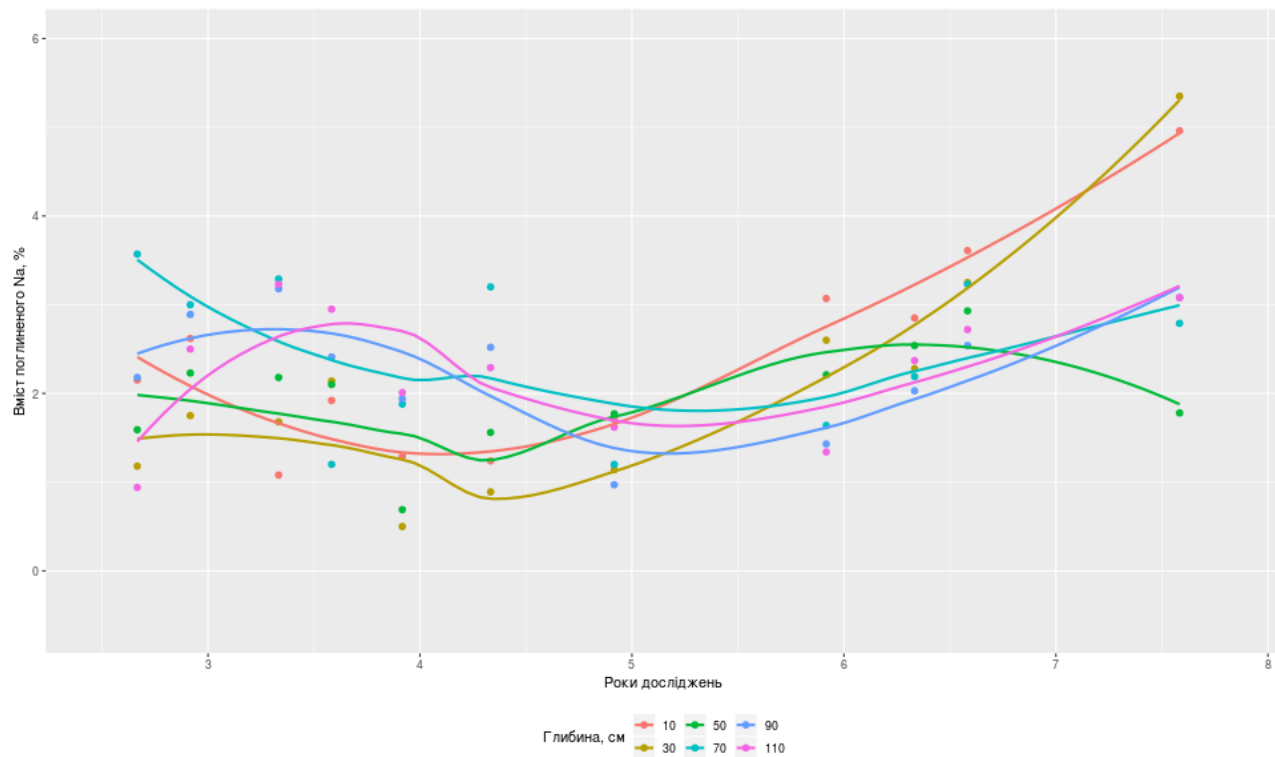
Щоб дізнатись які пакети встановлені в системі потрібно набрати:

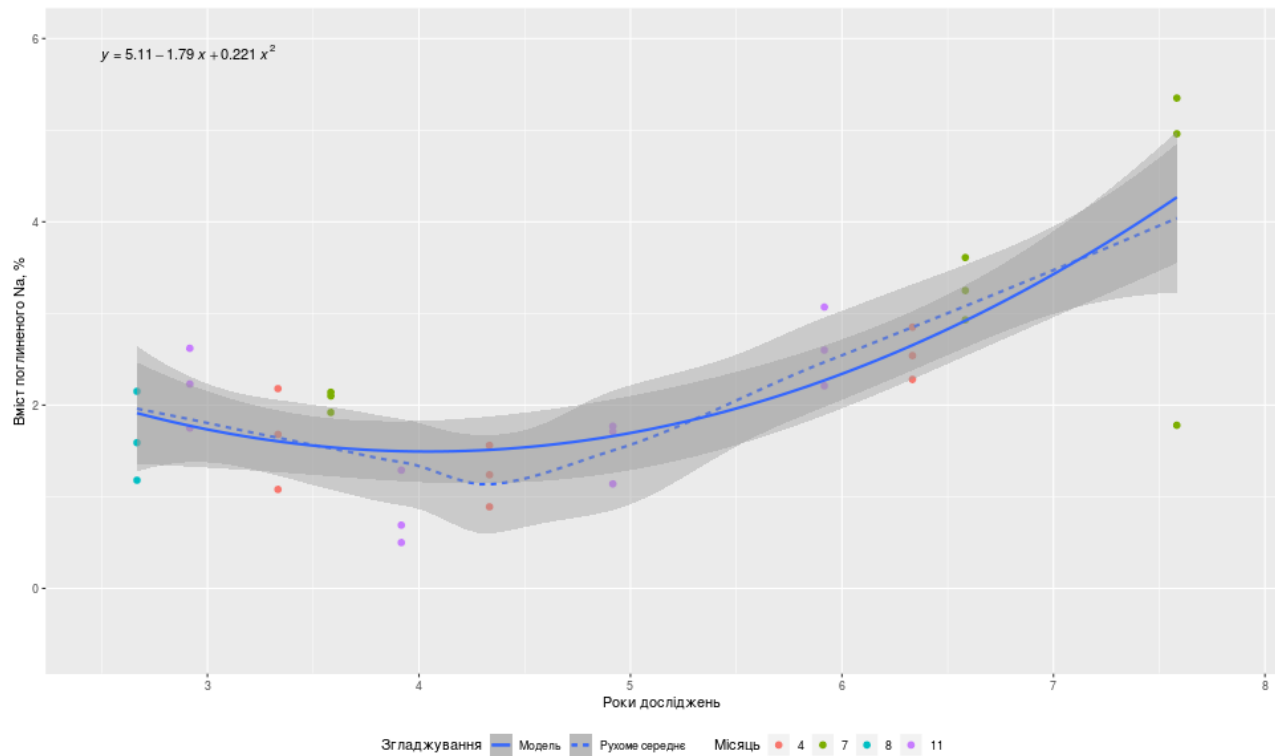
```
library()
```

Встановлення і поновлення пакетів виконується командами `install.packages()` і `update.packages()`.

Особливо слід відзначити бібліотеку `ggplot2`, яка дозволяє створювати досить складні графіки. Наприклад:



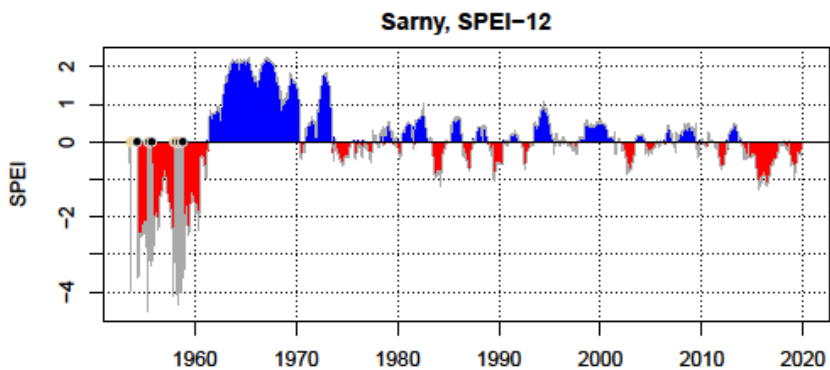
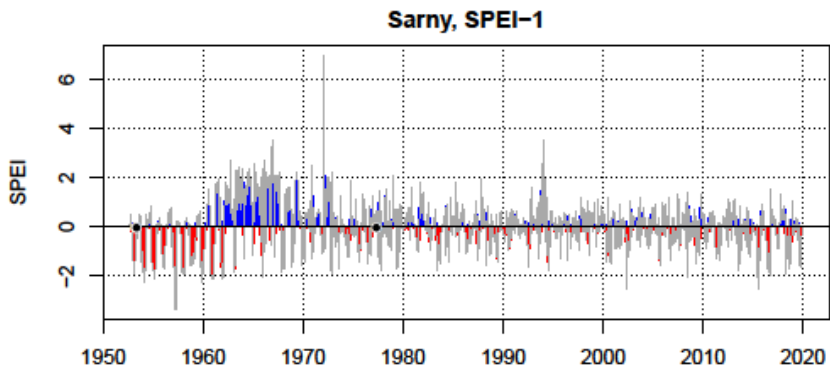




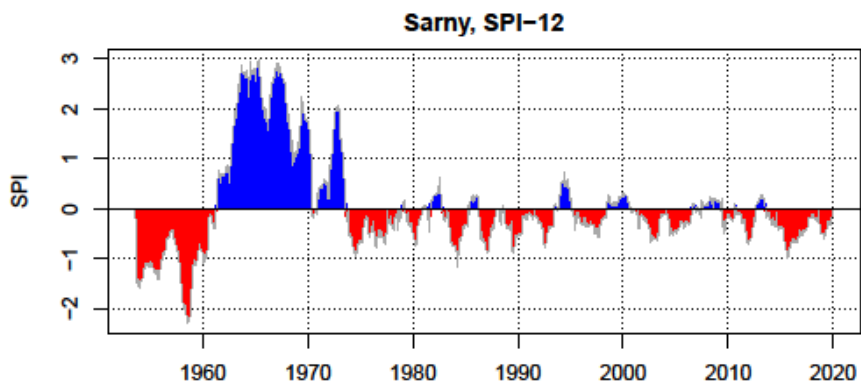
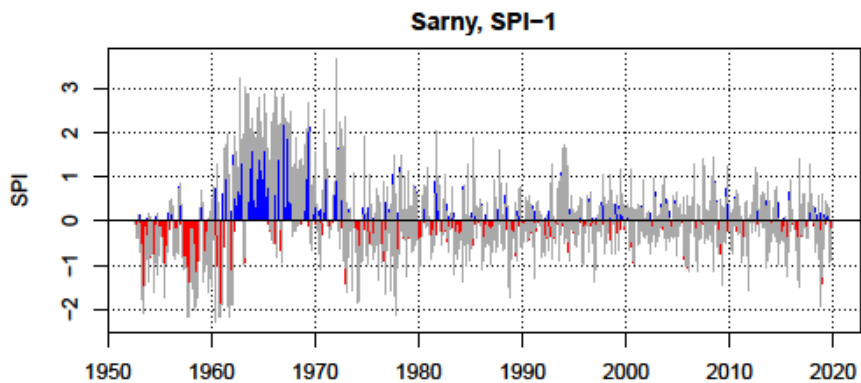
4. Графіка і графічні параметри

Наведемо приклад програми на мові R для розрахунки стандартизованого індексу опадів та сумарного випаровування (SPEI) по метеостанції Сарни.

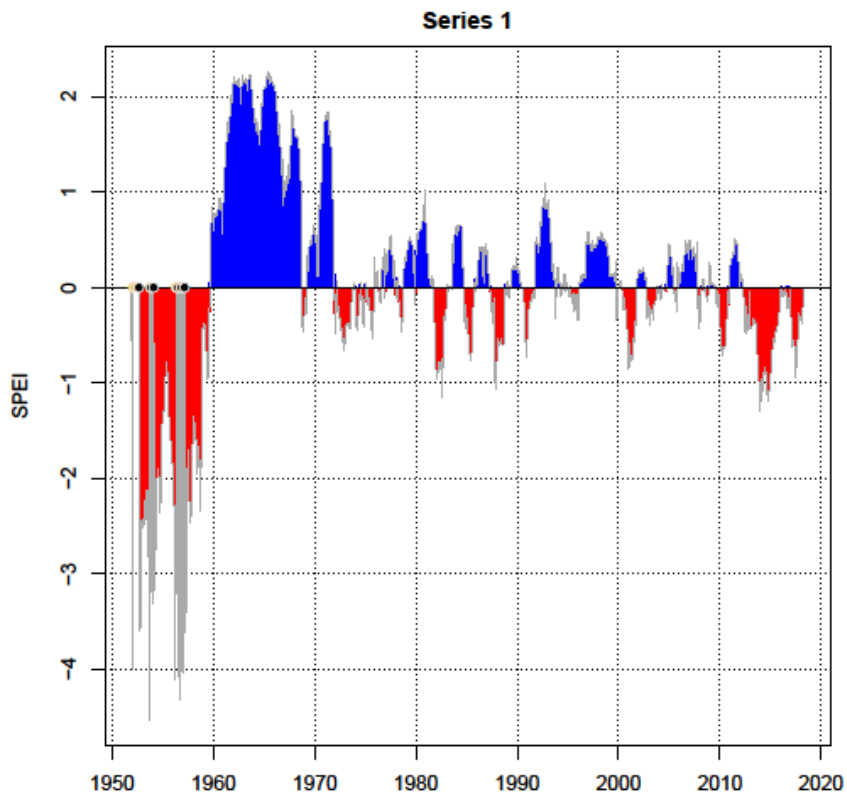
```
library(SPEI)
library(ggplot2)
Sarny <- read.table("Sarny.csv", sep=";", head=TRUE)
Sarny$PET <- thornthwaite(Sarny$TMED, 50.9450)
Sarny$BAL <- Sarny$PRCP-Sarny$PET
Sarny <- ts(Sarny[, -c(1,2)], end=c(2019,11), frequency=12)
spei1 <- spei(Sarny[, 'BAL'], 1)
spei12 <- spei(Sarny[, 'BAL'], 12)
class(spei1)
summary(spei1)
names(spei1)
spei1$call
spei1$fitted
spei1$coefficients
par(mfrow=c(2,1))
plot(spei1, main='Sarny, SPEI-1')
plot(spei12, main='Sarny, SPEI-12')
```



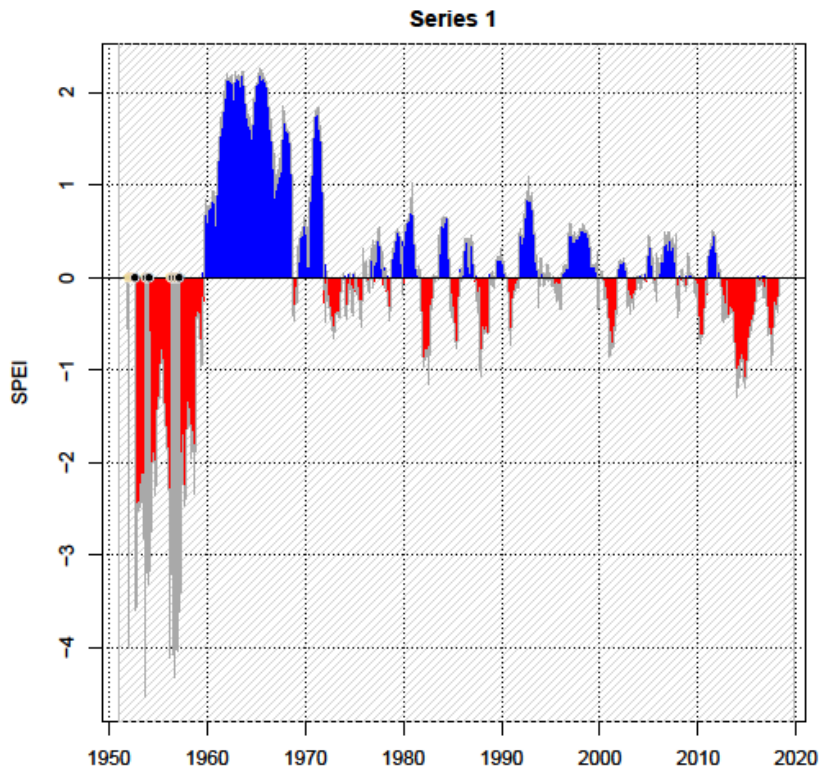
```
spi_1 <- spi(Sarny[, 'PRCP'], 1)
spi_12 <- spi(Sarny[, 'PRCP'], 12)
par(mfrow=c(2,1))
plot(spi_1, 'Sarny, SPI-1')
plot(spi_12, 'Sarny, SPI-12')
```

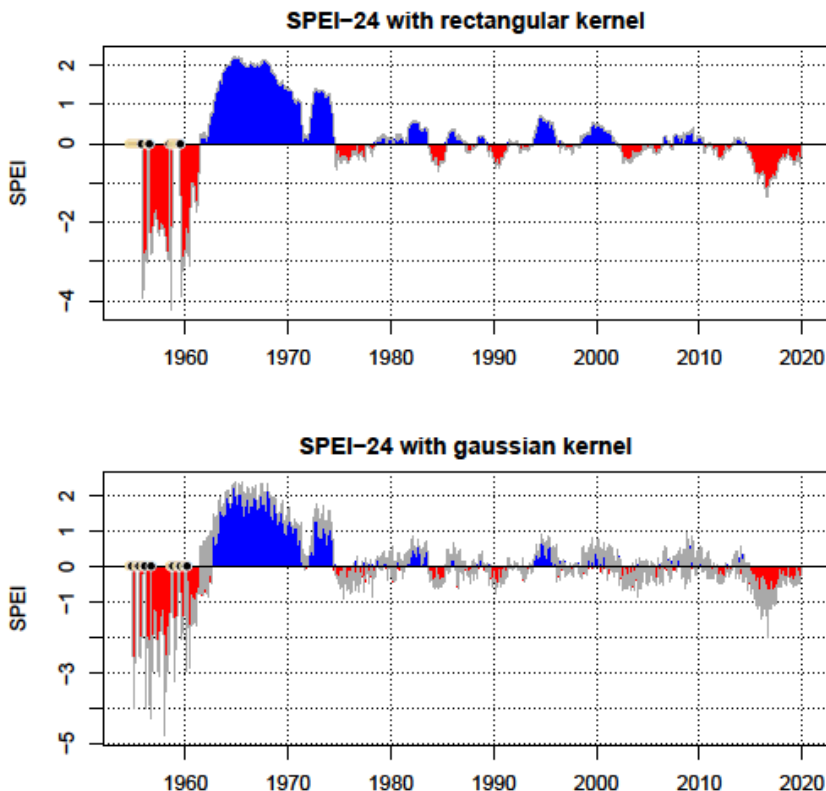
```
par(mfrow=c(1,1))  
plot(spei(ts(Sarny[, 'BAL'], freq=12, start=c(1951,1))), 12))
```



```
plot(spei(ts(Sarny['BAL'], freq=12, start=c(1951,1)), 12, ref.start=c(1951,1),  
ref.end=c(2019,11)))
```



```
spei24 <- spei(Sarny[, 'BAL'], 24)
spei24_gau <- spei(Sarny[, 'BAL'], 24, kernel=list(type='gaussian', shift=0))
par(mfrow=c(2,1))
plot(spei24, main='SPEI-24 with rectangular kernel')
plot(spei24_gau, main='SPEI-24 with gaussian kernel')
```



Як видно з прикладу графіки хорошої якості.

5. Доступ до віддалених баз даних водогосподарського призначення

Розроблений в 2015 році новий пакет WaterML R, надає доступ до даних інформаційної системи HydroServer Консорціуму університетів з розвитку гідрологічної науки (CUAHSI), як засобу для зберігання та управління даними. CUAHSI об'єднує більше 130 університетів США.

Отримана система дозволяє науковцям-дослідникам користуватися статистичним обчислювальним середовищем R разом з програмним забезпеченням з відкритим вихідним кодом HydroServer (для архівування та обміну даними). Користуючись пакетом WaterML R, користувач може отримати та проаналізувати дані від HydroServers з декількох організацій, які

перелічені в каталозі центру обробки даних CUAHSI та у глобальному каталозі даних Всесвітньої Метеорологічної Організації.

WaterML - це технічна норма і інформаційна модель, яка використовується для представлення гідрологічних структур часових рядів. Поточна версія WaterML 2.0.

Версія 1.0 WaterML була опублікована в 2009 році, є форматом обміну XML, розробленим спеціально для використання в США.

WaterML 2.0 - це відкритий стандарт, який гармонізує різні формати з різних організацій і країн, включаючи австралійський формат передачі водних даних, WaterML 1.0 з США, XHydro з Німеччини, і з існуючими форматами Open Geospatial Consortium (OGC). WaterML 2.0 був прийнятий як офіційний стандарт OGC у вересні 2012 року, схвалений Федеральним комітетом географічних даних США, і був запропонований для прийняття Світовою метеорологічною організацією World Meteorological Organisation (WMO).

Приклади використання:

- обмін даними для оперативних програм гідрологічного моніторингу;
- підтримка функціонування інфраструктури (наприклад, греблі, системи постачання);
- транскордонний обмін даними спостережень;
- випуск даних для публічного розповсюдження;
- посилення управління ліквідацією наслідків катастроф шляхом обміну даними;
- та обмін на підтримку національної звітності.

Стандарт був розроблений через процес гармонізації членами спільної робочої групи OGC-WMO Hydrology Domain.

Структура WaterML 2.0 використовує існуючі стандарти OGC, насамперед спостереження та вимірювання (O&M) та географічну мову розмітки (GML). Це підвищує узгодженість та сумісність з іншими стандартами та веб-службами. Використовуючи стандарт O&M, WaterML 2.0 визначає типи, що дозволяють стандартне визначення основних властивостей, що відносяться до гідрологічних часових рядів, включаючи:

- Спостережуване явище
- Просторовий контекст
- Тимчасові межі
- Процедура, що використовується для генерації часових рядів (наприклад, вихідні дані з датчика)
- Метадані для конкретних результатів, такі як класифікатори часових рядів, типи інтерполяції, коментарі, коди якості тощо
- Точки моніторингу
- Колекції суміжних об'єктів
- Словники для специфічних для домену термінів

Основна інформаційна модель визначається за допомогою уніфікованої мови моделювання, що забезпечує гнучкість у створенні специфічних для реалізації кодувань. Стандарт визначає відповідність XML- кодування GML, що дозволяє використовувати з OGC Web Services.

Служба спостереження за датчиками Sensor Observation Service (SOS) - це веб-служба для запиту даних датчиків у реальному часі та часових рядів даних датчиків і є частиною веб-датчика. Запропоновані дані датчиків містять описи самих датчиків, кодованих у мові моделі датчика (SensorML), і виміряні значення у форматі кодування спостережень та вимірювань (O&M). Веб-сервіс, як і обидва формати файлів, є відкритими стандартами і специфікаціями того ж імені, які визначені Консорціумом відкритих геопросторових об'єктів (OGC).

Якщо SOS підтримує транзакційний профіль (SOS-T), нові датчики можуть бути зареєстровані в інтерфейсі сервісу, а вимірювальні значення - вставлені. Реалізація SOS може використовуватися як для даних з датчиків in-situ, так і дистанційного зондування. Крім того, датчики можуть бути мобільними або стаціонарними.

З 2007 року SOS є офіційним стандартом OGC. Перевагою SOS є те, що дані датчиків - будь-якого роду - доступні в стандартизованому форматі з використанням стандартизованих операцій. Таким чином, веб-доступ до даних датчиків спрощується. SOS також дозволяє легко інтегруватися в існуючі інфраструктури просторових даних або географічні інформаційні системи.

У 2016 році OGC затвердила стандартну специфікацію API SensorThings, новий RESTful і JSON-стандарт надають функції, подібні до SOS. Оскільки обидва API SensorThings і SOS базуються на OGC / ISO19156: 2011, ці дві специфікації були продемонстровані в OGC IoT пілоті, що вони можуть взаємодіяти один з одним. SOS має три так звані основні операції, які повинні бути забезпечені кожною реалізацією.

Операція GetCapabilities дозволяє запитувати службу для опису інтерфейсу послуги та доступних даних датчиків. Для використання SOS функція GetObservation, ймовірно, є найбільш важливою. Він може бути використаний для отримання даних для конкретних датчиків.

Функція DescribeSensor повертає детальну інформацію про датчик або систему датчиків і процесах виробництва. Основні операції (основний профіль) GetCapabilities повертає опис служби XML з інформацією про інтерфейс (запропоновані операції та кінцеві точки), а також доступні дані датчиків, такі як період, для якого доступні дані датчиків, сенсори, які виробляють виміряні значення, або явища, які спостерігаються (для наприклад температура повітря).

GetObservation дозволяє виконувати запити на основі спостережуваних значень, включаючи метадані. Виміряні значення та їх метадані повертаються у форматі спостережень та вимірювань (O&M).

DescribeSensor - забезпечує метадані датчика в SensorML. Опис датчика може містити інформацію про датчик в цілому, ідентифікатор і класифікацію, положення і спостережувані явища, а також деталі, такі як дані калібрування.

Транзакційні операції (транзакційний профіль) RegisterSensor дозволяє реєструвати новий датчик в розгорнутому SOS.

InsertObservation можна використовувати для вставки даних для вже зареєстрованих датчиків у SOS.

Розширені операції (розширений профіль) GetResult надає можливість запитувати покази сенсорів без метаданих, що відповідають узгодженим метаданим (наприклад, датчик, спостережуваний об'єкт).

GetFeatureOfInterest повертає геооб'єкт, властивості якого контролюються датчиками в кодуванні мови географічної розмітки. GetFeatureOfInterestTime надає періоди часу, в яких доступні вимірювання спостережуваного об'єкта в SOS.

DescribeFeatureType повертає тип спостережуваних геооб'єктів (XML Schema). DescribeObservationType повертає тип спостереження (XML Schema), наприклад, om: Measurement).

GetObservationById дозволяє запитувати конкретне спостереження, використовуючи ідентифікатор, повернутий службою як відповідь на операцію InsertObservation.

DescribeResultModel надає XML-схему вимірюваного значення, що особливо важливо для складних вимірювань, таких як багатоспектральні дані.

Розглянемо приклад доступу до даних CUAHSI з мови програмування R та з використанням пакетів WaterML та ggplot2:

```
##Встановлення пакетів WaterML та ggplot2
```

```
install.packages("WaterML")
```

```
install.packages("ggplot2")
```

```
##Завантаження пакетів WaterML та ggplot2
```

```
library(WaterML)
```

```
library(ggplot2)
```

```
##Встановлення адреси сервера даних
```

```
USGS <- "http://hydroportal.cuahsi.org/nwisdv/cuahsi_1_1.asmx?WSDL"
```

```
##Завантаження потрібних даних для аналізу
```

```
AirTemp<-
```

```
GetValues(USGS,"NWISDV:01104480","NWISDV:00020DataType=MEAN",startDate = "2014-09-01", endDate="2014-09-30")
```

```
##Візуалізація отриманих даних
```

```
View(AirTemp)
```

```
##Побудова простого графіка стандартними засобами мови R  
plot(AirTemp$time,AirTemp$DataValue)
```

```
##Відображення загальної статистики отриманих даних
```

```
summary(AirTemp$DataValue)
```

```
##Створення графіка з використанням пакету ggplot  
g <- ggplot(AirTemp, aes(x = AirTemp$time, y = AirTemp$DataValue)) +  
geom_point()
```

```
##Відображення кольорового графіка (залежність кольору точок від величини  
температури)
```

```
g + geom_point(aes(color = AirTemp$DataValue))
```

Література: 3, 5.

Рекомендована література

1. Наглядная статистика. Используем R! А.Б. Шипунов, Е.М. Балдин, П.А. Волкова и др. 2014. 296 с.

URL: <https://cran.r-project.org/doc/contrib/Shipunov-rbook.pdf>

2. Data Visualisation with R. 100 Examples / Thomas Rahlf . Springer: 2017. 390 p.

URL: https://www.amazon.com/Data-Visualisation-R-100-Examples/dp/3319497502/ref=sr_1_1?s=books&ie=UTF8&qid=1518781154&sr=1-1&keywords=1.+data+visualization+with+r.+100+examples

3. ggplot2 Elegant Graphics for Data Analysis. Hadley Wickham. Springer: 2016. – 268 p.

URL: https://www.amazon.com/ggplot2-Elegant-Graphics-Data-Analysis/dp/331924275X/ref=sr_1_1?s=books&ie=UTF8&qid=1518781284&sr=1-1&keywords=ggplot2+Elegant+Graphics+for+Data+Analysis.+Hadley+Wickham

4. Modern Data Science with R / Benjamin S. Baumer. CRC Press: 2017. - 557p.URL:https://www.amazon.com/Modern-Science-Chapman-Texts-Statistical/dp/1498724485/ref=sr_1_1?s=books&ie=UTF8&qid=1518781435&sr=1-1&keywords=Modern+Data+Science+with+R+%2F+Benjamin+S.+Baumer

5. Сайт розробників мови програмування R, програмного середовища для статистичних обчислень, аналізу та представлення даних в графічному вигляді. URL: <https://cran.r-project.org/>